IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:				
	Yves Gattegno	§	Confirmation No.	6443
		§		
Serial No.:	10/593,262	§	Group Art Unit:	2128
		§		
Filed:	September 18, 2006	§	Examiner:	Patel, Shambhavi K.
		§		
For:	Method for the Software Emulation of the Hard Disks of a Computer Platform at the Operating System Thereof, With On-the-fly Parameter-adaptive Management of Read and Write Requests	§	Atty Docket:	GATTEGNO 1 200800984-6 HPQB:0195

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF TRANSMISSION OR MAILING
37 C.F.R. 1.8

I hereby certify that this correspondence is being transmitted by facsimile to the United States Patent and Trademark Office in accordance with 37 C.F.R. § 1.6(d), or is being transmitted via the Office electronic filing system in accordance with 37 C.F.R. § 1.6(a)(4), or is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date below:

February 17, 2010	/Nathan E. Stacy/
Date	Nathan E. Stacy

APPEAL BRIEF PURSUANT TO 37 C.F.R. §§ 41.31 AND 41.37

This Appeal Brief is being filed in response to the Final Office Action mailed on July 22, 2009, and in furtherance of a Notice of Appeal filed October 22, 2009. In the associated documents, the Appellant has requested a two-month extension for this Appeal Brief under 37 C.F.R. § 1.136(a). Therefore, this Appeal Brief is hereby timely filed on February 17, 2010.

1. **REAL PARTY IN INTEREST**

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 11445 Compaq Center Dr. W, Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

2. **RELATED APPEALS AND INTERFERENCES**

The Appellant is unaware of any other appeals or interferences related to this Appeal. The undersigned is the Appellant's legal representative in this Appeal.

3. **STATUS OF CLAIMS**

Claims 1-46 are currently pending, are currently under rejection and, thus, are the subject of this appeal.

4. **STATUS OF AMENDMENTS**

There are no outstanding amendments to be considered by the Board.

5. **SUMMARY OF CLAIMED SUBJECT MATTER**

The Application contains one independent claim, namely, claim 1, which is the subject of this Appeal. In this Appeal Brief, the line number citations are provided to the English translation of the PCT Application as filed on September 18, 2006.

As an example, independent claim 1 relates generally to a method which "totally emulates the software of hard disks at the level of the data blocks, also called sectors, or at the level of the file system, therefore permitting the use of file systems accepted by the operating system in emulated hard disks of any type." *See* Application, p. 3, ll. 4-7; *see also*, p. 1, ll. 7-14. The application also contains dependent claims 2-46. The subject matter of claim 1 is summarized below.

With regard to independent claim 1, discussions of the recited features can be found at least in the below-cited locations of the specification and drawings. By way of example, claim 1 recites a method that is performed by a suitably programmed processor. *See, e.g.*, p. 1, l. 19-p. 2, l. 2. The method provides for the software emulation of hard disks of a data processing platform at the level of an operating system with parameterizable management of requests for writing and reading data. *See id.* at p. 1, ll. 15-18; p. 4, ll. 19-23; p. 4, l. 28-p. 5, l. 2; p. 7, ll. 15-23; p. 8, ll. 12-13; p. 9, ll. 19-23; p. 9, l. 26-p. 10, l. 11; p. 14, ll. 16-19. The method consists of creating a representation of a real hard disk, wherein the sequence and location for loading and execution of components of the operating system of the data processing platform may be modified. *See id.* at p. 3, ll. 9-11; p. 13, ll. 1-7; p. 13, ll.12-18. The method also includes loading on said data processing platform one or more peripheral drivers, wherein at least one of the peripheral drivers communicates with a data storage support containing the data of the representation of the real hard disk. *See id.* at p. 3, ll. 11-14; p. 3, ll. 16-19; p. 4, ll. 16-19; p. 4, ll. 26-28; p. 7, ll. 3-6; p. 7; ll. 8-12; p. 11, l. 27-p. 12, l. 3. Further, the method includes simulating behavior of the real hard disk for the operating system, wherein the method transforms programming contained on the real hard disk into an emulated hard disk capable of controlling read and write operations on a client station and among two or more client stations. *See id.* at p. 3, ll. 13-15; p. 4, ll. 1-7; p. 4, ll. 17-19; p. 4, ll. 26-28; p. 6, ll. 3-6; p. 11, ll. 8-11; p. 11, ll. 12-26.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. First Ground of Rejection for Review on Appeal

The Appellant respectfully urges the Board to review and reverse the Examiner's first ground of rejection in which the Examiner rejected claims 1-6, 11-29, and 38-46 under 35 U.S.C. § 103(a) as being unpatentable over Flouris, Michail D., et al., "The Network RamDisk: Using Remote Memory on Heterogeneous NOWs," Cluster Computing 2, pp. 281-293 (1999), (hereinafter "Flouris") in view of U.S. Patent No. 5,991,542 to Han, et al. (hereinafter "Han").

B. Second Ground of Rejection for Review on Appeal

The Appellant respectfully urges the Board to review and reverse the Examiner's second ground of rejection in which the Examiner rejected claims 7-10 and 30-37 under 35 U.S.C. § 103(a) as being unpatentable over Flouris in view of Han in view of U.S. Patent Application Publication No. 2003/0208675 by Burokas, et al. (hereinafter "Burokas").

7. ARGUMENT

As discussed in detail below, the Examiner has improperly rejected the pending claims. Further, the Examiner has misapplied long-standing and binding legal precedents and principles in rejecting the claims under 35 U.S.C. § 103(a). Accordingly, the Appellant respectfully requests full and favorable consideration by the Board, as the Appellant asserts that claims 1-46 are currently in condition for allowance.

A. Ground of Rejection No. 1

With respect to the rejection of claims 1-6, 11-29, and 38-46 under 35 U.S.C. § 103(a) as being unpatentable over Flouris in view of Han, the Examiner stated that:

> **Flouris discloses** a method, performed by a suitably programmed computer, for software emulation of hard disks of a data processing platform at the level of the operating system with parameterizable management of requests for writing and reading data consisting in:
>
> a. creating a representation of a real hard disk **(page 10: NRD client asks as a normal disk)**wherein the sequence and location for loading and execution of components of the operating system of the data processing platform may be modified **(section 4.1.6: filenames ordered in random permutations)**
>
> b. loading on said data processing platform one or more peripheral drivers **(section 3.1: NRD client is a disk device driver)**, wherein at least one of the peripheral drivers communicates with

> a data storage support containing the data of the representation of the real hard disk **(section 2.1: once the driver is mounted the RamDisk operates as a regular disk)**.
>
> c. simulating the behavior of the real hard disk for the operating system **(section 4: performance of RamDisk measured by running the software)**
>
> **Flouris does not explicitly disclose** transforming programming contained on the real hard disk into an emulated hard disk. **Han teaches** using disk images to emulate storage volumes **(abstract)** wherein programming contained on the real hard disk in transformed into an emulated hard disk **(abstract: image of a data storage volume is stored in a file)** wherein the emulated hard disk is capable of controlling read and write operations **(column 5 lines42-45: read and write requests)** on a client station and among two or more client stations **(column 7 lines 58-50: client computers)**. At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Flouris and Han because the use of said image files allows for independence from format requirements, the ability to compress the data, and end-to-end verification **(Han: column 2 lines 33-53)**.

Final Office Action, p. 3. (Emphasis in original). The Appellant respectfully traverses this rejection.

The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (B.P.A.I. 1979). To *establish prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka, 180 U.S.P.Q.* 580 (C.C.P.A. 1974). Although a showing of obviousness under 35 U.S.C. § 103 does not require an express teaching, suggestion or motivation to combine prior art references, such a showing has been described by the Federal Circuit as providing a "helpful insight" into the obviousness inquiry. *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350, 550 U.S. 398, 82 U.S.P.Q.2d 1385 (2007). Moreover, obviousness cannot be established by a mere showing that each claimed element is present in the prior art. *Id.* The Examiner must

cite a compelling reason why a person having ordinary skill in the art would combine known elements in order to support a proper rejection under 35 U.S.C. § 103. *Id.*

Further, it is improper to combine references where the references teach away from their combination. *In re Grasselli*, 713 F.2d 731, 743, 218 U.S.P.Q. 769, 779 (Fed. Cir. 1983); M.P.E.P. § 2145. Moreover, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti*, 270 F.2d 810, 123 U.S.P.Q. 349 (CCPA 1959); see M.P.E.P. § 2143.01.

Flouris and Han do not disclose all of the elements of claim 1, alone or in any hypothetical combinations.

Flouris is directed to a disk driver that is intended to reduce latency of data storage operations (reads and writes) by emulating a disk drive in the idle main memory of two or more networked computers or workstations rather than using "traditional magnetic disks." See Flouris, p. 2. Flouris accomplishes the reduction in latency by running a NRD (Network RamDrive) server process on each workstation with memory blocks that may be used for storage. *See id.* at p. 282, § 2.2. A system wishing to utilize the storage, e.g., a NRD client, has a disk driver device called an "NRD client process" that handles all read and write processes. *See id.* at p. 287, § 3.1. The NRD client process searches for idle main memory blocks among the networked workstations. *See id.* The NRD client process then stores two maps in its memory, a "block table" with the locations of each available block on the networked servers and a bitmap with an indication as to whether each block is full or empty. *See id.* After the maps are built, the NRD client functions as a normal disk drive, accepting block I/O requests from the computer operating the NRD client. *See id.*

However, Flouris does not disclose that "the *sequence and location* for loading and execution of components of the operating system of the data processing platform *may be modified*," as recited by claim 1. (Emphasis added). The specification states that the

"orders of priority depend on the sequence previously followed for the redirections of data write requests. In this way it is assured, from the standpoint of the client station, that the emulated hard disk is always coherent." Specification, p. 8, ll. 24. In contrast, the cited section of Flouris does not disclose either a *sequence or loading location* at all, but a *random permutation* of filenames that may be accessed to test the speed of reading/writing to the NRD. *See* Flouris, p. 298, § 4.1.6. Furthermore, Flouris does not disclose that any of these files may be "components of the operating system," as recited in claim 1.

Han does not remedy the deficiencies of Flouris, alone or in any combination with Flouris. Han is directed to a method for distributing computer software by building an image of a data storage volume in a file that has a format allowing it to exhibit a behavior that is the same as the storage volume itself. *See* Han, col. 2, ll. 32-38. The data in the image file can be compressed to take up less space. *See id.*, col. 2, ll. 42-45. The images are used for transmission or downloading of the files to facilitate installation of software on remote computers. *See id.*, col. 3, ll. 17-21. However, Han does not disclose that the image contains components of the operating system, much less that the sequence and location for loading and execution may be modified, as recited in claim 1. Thus, for this reason, the Appellant respectfully asserts that claim 1 is allowable over Flouris and Han, alone or in any combination.

Further, nothing in Han or Flouris discloses that the method "transforms programming contained on the real hard disk into *an emulated hard disk capable of controlling read and write operations* on a client station and among two or more client stations." as recited by claim 1. (Emphasis added). The Examiner admits that Flouris does not disclose this element, but asserts that it is disclosed in Han, stating that "the emulated hard disk is capable of controlling read and write operations (**column 5 lines42-45: read and write requests**) on a client station and among two or more client stations (**column 7 lines 58-50: client computers**)." *See* Final Office Action, p. 3 (emphasis in original).

However, the Appellant respectfully disagrees with this characterization of Han. At col. 7, lines 42-45, Han merely states that "[i]n a similar manner, when a disk image is mounted, a driver is employed to carry out the task of providing data to the image file and retrieving data therefrom." Nothing in Han discloses that the driver is located in the emulated hard disk. In fact, Han specifically notes that each of the storage devices that may be located in a computer "has an associated driver 32-38 stored *in the memory of the computer*." Han, col. 5, ll. 31-33 (emphasis added). In fact, one of ordinary skill in the art would recognize that the system in Han would not be able to access the emulated storage device unless the driver is already present in the memory of the computer. Further, the Appellant notes that Han states that "the installation at each remote computer 68 is *controlled by the individual client computers themselves*, and therefore does not require the resources of the network server 62." Han, col. 7, ll. 49-52 (emphasis added). Further, nothing in Han discloses that the emulated hard drive itself controls read and write operations. Thus, neither Han nor Flouris discloses that the emulated hard drive is able to control read and write operations. Accordingly, independent claim 1 is patentable over Han and Flouris for at least this additional reason.

Finally, claim 1 recites that the software emulation has "parameterizable management of requests for writing and reading data." Although this element is in the preamble, the preamble of claim 1 recites specific elements of the claim (e.g., "an operating system") and, further, "is necessary to give life, meaning, and vitality to the claim." *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999); *see also* M.P.E.P. § 2111.02. Thus, the element should be construed as if in the balance of the claim.

The parameterizable management of requests for reading and writing data is discussed in several places in the specification. For example, the specification states that "[t]he possibility exists on the level of said service of managing the written data in parameterizable form and of being able, in particular, to provide in a storage space, specific to the client station/virtual hard disk pair, storage space that is not the common

storage space of all clients." Specification, p. 14, ll. 16-19. Further, the Specification notes that "[t]he volatility or persistence of the storage space used for the storage of the written data is parameterizable when this would make sense." *Id.* at p. 8, ll. 12-13. Neither Flouris nor Han discloses parameterization management of requests for reading and writing data as recited in claim 1. Accordingly, independent claim 1 is patentable over Han and Flouris for at least this additional reason.

For at least the reasons discussed above, the cited references, whether alone or in any hypothetical combination, fail to disclose all of the elements recited in independent claim 1. Accordingly, this claim is allowable over Flouris and Han, alone or in any hypothetical combination. For at least the same reasons, its dependent claims 2-46 are allowable over the cited references. Accordingly, the Appellant respectfully requests the Board to reverse the rejection of claims 1-46 under 35 U.S.C. § 103(a).

B. Ground of Rejection No. 2

The Appellant respectfully urges the Board to review and reverse the Examiner's second ground of rejection in which the Examiner rejected claims 7-10 and 30-37 under 35 U.S.C. § 103(a) as being unpatentable over Flouris in view of Han in view of Burokas. Claims 7-10 and 30-37 depend from claim 1 and, thus, are allowable over Han and Flouris for at least the reasons discussed above. Therefore, these claims stand or fall with claim 1.

Further, Burokas does not remedy the deficiencies of Flouris and Han with respect to independent claim 1, alone or in any combinations with these references. Burokas discloses a method for booting computers from an emulated drive on a network server after the computers are resumed from hibernation. Burokas, paras. [0020], [0028]. However, Burokas does not disclose that "the *sequence and location* for loading and execution of components of the operating system of the data processing platform *may be modified*," as recited by claim 1. (Emphasis added). In contrast, the method of Burokas "allows all client PCs to boot the same hibernation image. This translates to easier

maintenance since each client is no longer running unique copies of the O/S or the applications." *Id.*, para. [0013].

Further, Burokas does not disclose that the method "transforms programming contained on the real hard disk into an emulated hard disk capable of controlling read and write operations on a client station and among two or more client stations," as recited by claim 1. In contrast, Burokas is directed to a method for booting a PC, which may then run it own code. *Id.*, para. [0009]. The emulation code is not on a separate drive, but is contained within the client PC (such as in a NIC), which accesses code on the network server. Burokas, paras. [0011]. Although Burokas does disclose a boot image that can be used to boot a computer from hibernation, Burokas does not disclose a method that "transforms programming contained on the real hard disk into an emulated hard disk," e.g., by copying information from an operational hard disk to a virtual disk.

In addition to the reasons discussed above, Burokas does not disclose the "parameterization management of requests for reading and writing data" as recited in claim 1. Accordingly, independent claim 1 is patentable over any hypothetical combination of Burokas with Flouris or Han for at least this additional reason.

For at least the reasons discussed above, none of the cited references, alone or in any hypothetical combination, disclose all of the elements recited in independent claim 1. Accordingly, this claim is allowable over Han, Flouris, and Burokas, alone or in any hypothetical combinations. For at least the same reasons, their respective dependent claims 7-10 and 30-37 are allowable over the cited references. Accordingly, the Appellant respectfully requests the Board to reverse the rejection of claims 7-10 and 30-37 under 35 U.S.C. § 103(a).

C. Request for Reversal of the Rejections

In view of the reasons set forth above, the Appellant respectfully requests the Board to reverse the rejections of claims 1-46 under 35 U.S.C. § 103(a).

Conclusion

The Appellant respectfully submits that all pending claims are in condition for allowance. However, if the Examiner or Board wishes to resolve any other issues by way of a telephone conference, the Examiner or Board is kindly invited to contact the undersigned attorney at the telephone number indicated below.

Respectfully submitted,

Date: February 17, 2010

/Nathan E. Stacy/
Nathan E. Stacy
Reg. No. 52,249
International IP Law Group, P.C.
(832) 375-0200

CORRESPONDENCE ADDRESS:

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
Fort Collins, Colorado 80528

8. APPENDIX OF CLAIMS ON APPEAL

Listing of Claims:

1. Method, performed by a suitably programmed processor, for software emulation of hard disks of a data processing platform at the level of an operating system with parameterizable management of requests for writing and reading data, consisting of:

creating a representation of a real hard disk, wherein the sequence and location for loading and execution of components of the operating system of the data processing platform may be modified,

loading on said data processing platform one or more peripheral drivers, wherein at least one of the peripheral drivers communicates with a data storage support containing the data of the representation of the real hard disk, and

simulating behavior of the real hard disk for the operating system, wherein the method transforms programming contained on the real hard disk into an emulated hard disk capable of controlling read and write operations on a client station and among two or more client stations.

2. Method as claimed in claim 1, wherein the management of said data write requests that the operating system sends to the emulated hard disk is accomplished at the peripheral driver level and/or at the level of an optional hard disk server service on a data processing network, the written data being stored according to the parameterization of said peripheral drivers and/or said service server of the hard disk on the network

- either directly in the support containing the emulated hard disk,
- or in the memory, random access or virtual, accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the operating system using the emulated hard disk,
- or in a non-volatile storage space accessible to the operating system using the emulated hard disk,

- or in a volatile storage space accessible to the server service of emulated hard disks on a data processing network,
- or in a non-volatile storage space accessible to the server service of emulated hard disks on a data processing network.

3. Method as claimed in claim 1, wherein the management of the data reading requests that the operating system issues to the emulated hard disk is accomplished at the peripheral driver level and/or at the level of an optional hard disk server service on a data processing network, the readings of previously written data being performed in the storage space:

- either directly in the support containing the emulated hard disk,
- or in the random access or virtual memory accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the operating system using the emulated hard disk,
- or in a nonvolatile storage space accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the server service of emulated hard disks on a data processing network,
- or in a non-volatile storage space accessible to the server service of emulated hard disks on a data processing network.

4. Method as claimed in claim 1, wherein the emulation of the hard disk provided to the operating system of the client station is accomplished by the agency of a single, monolithic peripheral driver which communicates with the operating system in the manner of a hard disk and which communicates with the support containing the data of said emulated hard disk in a manner specific to this support.

5. Method as claimed in claim 1, wherein the data of the emulated hard disk or disks are accessible to the client stations via a data processing network.

6. Method as claimed in claim 1, wherein when an emulated hard disk is started up, the method further comprises providing a low level micro-software module to access the data contained in the emulated hard disk, wherein the operating system is started up at the client station.

7. Method as claimed in claim 6, wherein the data processing network comprises a plurality of client stations, the client stations using a bootup PROM, the method further comprising using the bootup PROM to control communications via the data processing network.

8. Method as claimed in claim 7, wherein the low level micro-software module is loaded in the memory of the client station and executed using the bootup PROM.

9. Method as claimed in claim 6, wherein the low level micro-software module is loaded in memory of the client station and executed as a component of the BIOS of the client station, said low level micro-software module providing the same functions as the access services on real hard disks provided by the BIOS.

10. Method as claimed in claim 6, wherein the low-level micro-software is loaded in memory of the client station from a third party data support supported as a startup peripheral by the client station.

11. Method as claimed in claim 5, wherein at least one peripheral driver loaded and executed by the operating system of the client station provides the functions of access, via the data processing network, to the data contained in the emulated hard disks.

12. Method as claimed in claim 1, wherein if the data support containing the data of the emulated hard disk(s) is a support that does not provide for writing in real time, or does not accept writing of data directly in the support containing the data of the emulated hard disk, the data writing requests issued by the operating system to the emulated hard

disk(s) are processed in such a way that the written data are stored in a storage space different from the data support containing the data of the emulated hard disk(s).

13. Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the random access memory of the client station.

14. Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the virtual memory of the client station.

15. Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in a data file accessible to the operating system of the client station.

16. Method as claimed in claim 1, wherein the data writing requests issued by the operating system to the emulated hard disk(s) are redirected to a single storage space, and wherein the storage space in which the written data are redirected may be changed during an operating session of the operating system of a client station.

17. Method as claimed in claim 12, wherein the storage space used for storage of the written data may be volatile or nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another.

18. Method as claimed in claim l6, wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of the client station.

19. Method as claimed in claim 1, wherein the data reading requests issued by the operating system are performed in different storage spaces during an operating session of the operating system of a client station.

20. Method as claimed in claim 19, wherein the data reading requests issued by the operating system to an emulated hard disk carried out in different storage spaces follow an order of priority.

21. Method as claimed in claim 5, wherein a server program is in charge at one of the client stations of the data processing network, on the one hand, of the communications via the data processing network with the client stations accessing the emulated hard disks, and on the other, of accessing the data support containing the data of the emulated hard disks.

22. Method as claimed in claim 21, wherein if the hard disk emulation system is parameterized so that the data write requests received by the server program are intended for a specific emulated hard disk they are not redirected but stored directly in a support containing the data of the emulated hard disk itself, and only one client station can access said emulated hard disk at a given time.

23. Method as claimed in claim 21, wherein in order to permit several client stations to access an emulated hard disk simultaneously, the server program is capable of redirecting specifically the data write requests issued by a client station A to a given storage space, and of redirecting the data write requests issued by another client station B to another given storage space.

24. Method as claimed in claim 1, wherein in order to permit the startup from and/or simultaneous access to the same emulated hard disk or 100% identical copies of the same emulated hard disk, components of the operating system loaded and executed by

the client stations or server program are capable of modifying, during or before their use by the operating system, data contained in the emulated hard disk.

25. Method as claimed in claim 1, wherein the emulation is performed for the operating system of the client stations at a level of a class of virtual peripherals of a file system type.

26. Method as claimed in claim 1, wherein the emulation is performed for the operating system of the client stations at the level of the class of disk peripherals itself and not at the file system level.

27. Method as claimed in claim 1, wherein data contained in the emulated hard disk are copied by a software tool executed at a client station from the real hard disk.

28. Method as claimed in claim 27, wherein the software tool creates an image directory that contains the data of the emulated hard disk.

29. Method as claimed in claim 27, wherein the software tool creates an image file that contains the data of the emulated hard disk.

30. Method as claimed in claim 1, wherein in order to permit startup from an emulated hard disk, the sequence of loading of the components of the operating system is modified so that all components of the operating system on which the peripheral drivers permitting access to the emulated hard disk depend are loaded and usable at the moment when the operating system accesses the emulated hard disk by using the peripheral drivers.

31. Method as claimed in claim 21, wherein in order to accelerate the simultaneous access by several client stations to the same emulated hard disk whose data

are contained in a data support accessible to a server station, the data are sent by the server station to the client stations using broadcast or multicast mechanisms.

32. Method as claimed in claim 31, wherein the data sent by broadcast or by multicast by the server station are stored by the client stations that accept them in a local cache situated in the memory of said client stations.

33. Method as claimed in claim 31, wherein a read request for data in the emulated hard disk issued by the operating system of a client station generates an data reading request sent to the server station only if said data are not already present in said local cache.

34. Method as claimed in claim 33, wherein the data read in the local cache are removed after being read by the client station so as to free up space in said local cache.

35. Method as claimed in claim 31, wherein the decision to send data by "multicast/broadcast" is made at the server module level which provides the functionalities necessary for the hard disk emulation at the client stations.

36. Method as claimed in claim 31, wherein the client stations may modify their subscription to receiving the data sent via "broadcast/multicast" by the server station.

37. Method as claimed in claim 32, wherein the client stations may erase the data from the local cache after a certain parameterizable time.

38. Method as claimed in claim 5, wherein data contained in the emulated the server module making the hard disks available to client stations may use any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

39. Method as claimed in claim 5, wherein a low level software program executed by the client stations permits access to the data contained in the emulated hard disks using any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

40. Method as claimed in claim 11, wherein the peripheral driver(s) executed by the client stations permit access to the data contained in the emulated hard disks any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

41. Method as claimed in claim 21, wherein if the data support containing the data of the emulated hard disk(s) is a support that does not provide writing in real time, or does not accept write operations directly in the support containing the data of the emulated hard disk, the server program providing the emulation of the hard disk at the client stations processes the data write requests issued by the operating system to the emulated hard disk(s) in such a way that the written data are stored in a storage space different from the data support containing the data of the emulated hard disk(s).

42. Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the random access memory of the server station.

43. Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the virtual memory of the server station.

44. Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in a data tile accessible to the server software.

45. Method as claimed in claim 21, wherein the storage space used for storage of the written data may be volatile or nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another.

46. Method as claimed in claim 16, wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of the client station.

9. **EVIDENCE APPENDIX**

None.

10. RELATED PROCEEDINGS APPENDIX

None.